

웹 프로그래밍

# 5장. 쿠키와 세션

동아대학교 컴퓨터공학과

# 쿠키와 세션의 필요성

- ❖ 로그인 정보, 온라인 쇼핑몰 장바구니 정보 등 인터넷 서비스에 연결이 지속되어야 하는 경우 사용
  - 즉, 사용자의 브라우저와 서버 간의 논리적인 연결
- ❖ 서버와 접속한 클라이언트의 정보를
  - 클라이언트가 가지고 있으면 **쿠키** **가**
    - ✓ 클라이언트에 대한 정보가 과자 부스러기처럼 남는다고 해서 쿠키라 불림
  - 서버가 가지고 있으면 **세션**
- ❖ 쿠키 혹은 세션 미제공 시, 매번 페이지 이동할 때마다 사용자의 정보를 계속 물어야 합니다.

1:1

ex)

- ❖ HTTP (Hyper Text Transfer Protocol): HTML 문서의 송수신을 위해 사용하는 프로토콜(프로토콜은 간단히 말해서 인터넷 통신 규약)
- ❖ 하지만 웹 서비스를 가능하게 하는 프로토콜 중 가장 핵심적인 HTTP 프로토콜은 비연결(connectionless) 지향형 통신 프로토콜
  - 웹 서버가 해당 페이지를 클라이언트에게 전송(응답)한 후 연결을 끊습니다.
  - 또 정보를 요청하면 서버는 이전 정보를 기억하지 않고, 새로운 연결을 만든 후 응답합니다.
  - 이러한 웹 서버는 이전에 전송한 정보를 전혀 기억하지 못하므로 무 상태(stateless) 서버라고 합니다.
- ❖ 쿠키 혹은 세션을 이용하여 비연결이라는 단점을 극복

# 쿠키(cookie)

- ❖ 쿠키는 서버가 아닌 클라이언트 컴퓨터의 하드디스크에 사용자의 정보를 저장하기 때문에 서버의 동작에 무리를 주지 않으면서 사용자의 정보를 적절하게 사용할 수 있습니다.
- ❖ 쿠키는 클라이언트에 저장된 적은 양의 정보입니다. 크기는 4KB 이하로 제한되어 있고 300개까지의 데이터 정보 배열을 저장할 수 있습니다.
  - 즉 쿠키를 사용할 수 있는 최대 용량은  $4\text{KB} * 300\text{개} = 1.2\text{MB}$
- ❖ iExplorer경우 C:\Users\사용자\AppData\Roaming\Microsoft\Windows\Cookies에 있는데, 탐색기로는 안보임
- ❖ 참고로 이미지 정보는 대부분 용량이 크기 때문에 경로만 저장

# 쿠키 사용 예제

(소스 안에 페이지지시자는 공간상 생략) p.238

## 01\_setCookies.jsp

```
<%  
    Cookie c = new Cookie("id", "james"); // 1. Cookie 객체 생성  
    c.setMaxAge(7 * 24 * 60 * 60);      // 2. 유효기간 설정(7일)  
    response.addCookie(c);              // 3. 클라이언트에 쿠키 전송  
  
    response.addCookie(new Cookie("pwd", "1234")); // 4. 유효기간설정 없으므로 브라우저 닫히면 사라짐  
    response.addCookie(new Cookie("age", "21"));  // 5. 이 쿠키도 마찬가지  
%>  
<h3>쿠키 설정 했다~</h3>
```

쿠키 설정 했다~

## 02\_getCookies.jsp

```
<h3> 클라이언트로부터 얻어온 Cookie </h3>  
<%  
    Cookie[] cookies = request.getCookies();  
    for (Cookie c : cookies) {  
        out.println(c.getName()+" : "+c.getValue()+"<br>");  
    }  
%>
```

클라이언트로부터 얻어온 **Cookie**

```
id : james  
pwd : 1234  
age : 21  
JSESSIONID : BDE4D00BC11BC2CA3CA5E19AEE409AB9
```

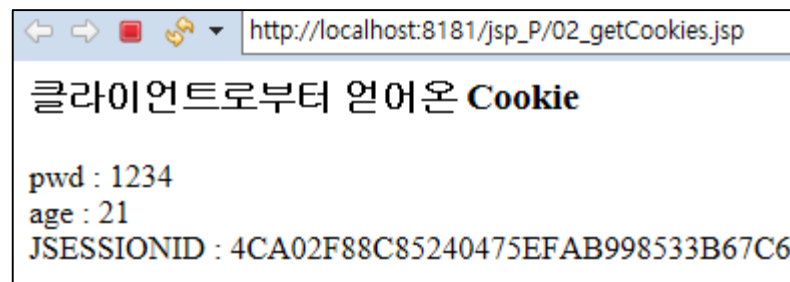
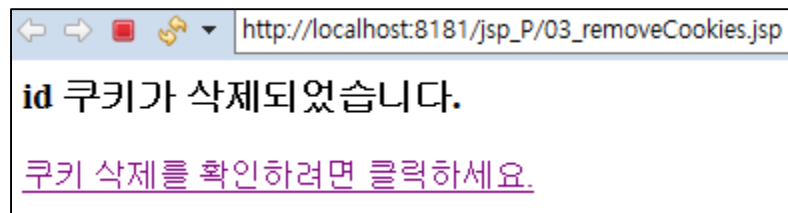
참고: 아직 세션을 배우지는 않았지만 쿠키의 정보 중 JSESSIONID에 대해서 의문을 가질 분들이 있을 것입니다. 세션 기술에서는 browser로 세션 ID를 보낼 때 쿠키 형태로 만들어서 전송하는데, JSESSIONID가 바로 세션 정보를 저장한 쿠키의 이름입니다.

# 쿠키 사용 예제

(소스 안에 페이지지시자는 공간상 생략) p.241

## 03\_removeCookies.jsp

```
<%  
    Cookie cookie = new Cookie("id", "");  
    cookie.setMaxAge(0); // 쿠키의 유효기간을 만료시킴  
    response.addCookie(cookie);  
%>  
<h3> id 쿠키가 삭제되었습니다. </h3>  
<a href="02_getCookies.jsp">  
    쿠키 삭제를 확인하려면 클릭하세요.  
</a>
```



- ❖ 쿠키의 단점: 쿠키가 저장되는 디렉토리만 뒤지면 얼마든지 정보를 알아낼 수 있으므로 PC 방과 같이 여러 사람이 PC를 공유하는 경우에는 개인의 정보가 유출될 수 있다는 우려가 있어 보안 유지가 되지 않는다는 단점이 있습니다.
- ❖ 또한 저장할 수 있는 데이터도 1.2MB로 한계가 있습니다.
- ❖ 세션을 사용하여 이러한 단점 극복
  - JSP(서버)에서만 접근 가능하므로 **보안 유지에 강력**합니다. 또한 저장할 수 있는 데이터에 한계가 없습니다.
  - 세션은 웹 브라우저를 닫기 전까지 페이지를 이동하더라도 사용자의 정보를 잃지 않고 서버에 보관할 수 있도록 하는 객체
  - 브라우저 단위당 한 개씩 존재

자주 쓰는 메소드	설명
Object <b>getAttribute</b> (String name)	이름에 해당되는 객체 값을 가져옵니다. (없을 경우에는 null을 반환) 반환값은 Object형이므로 반드시 형 변환을 하여 사용해야 합니다.
Enumeration <b>getAttributeNames</b> ()	세션에서 모든 객체들의 이름을 Enumeration 형으로 얻어 줍니다.
void <b>setAttribute</b> ( String name, Object value)	세션에 지정된 이름에 객체를 추가합니다.
void <b>setMaxInactiveInterval</b> ( int interval)	사용자가 다음 요청을 보낼 때까지 세션 유지하는 최대 시간(초 단위)을 설정합니다.
void <b>invalidate</b> ( )	해당 세션을 없애고 세션에 속해있는 값들을 없앱니다.



# 세션 설정 및 값 얻어오기

(jsp 소스 안에 페이지 지시자는 공간상 생략) p.245

## 04\_setSession.jsp

```
<%  
    session.setAttribute("id", "james");  
    session.setAttribute("pwd", "1234");  
    session.setAttribute("age", 21);  
%>  
<h3> 세션 설정했다~</h3>
```

세션 설정했다~

## 02\_getSession.jsp

```
<h3> 세션 값 얻어오기 </h3>  
<%  
    String id = (String) session.getAttribute("id");  
    String pwd = (String) session.getAttribute("pwd");  
    Integer age = (Integer) session.getAttribute("age");  
%>  
id : <%= id %> <br>  
pwd : <%= pwd %> <br>  
age : <%= age %> <br>
```

세션 값 얻어오기

id : james  
pwd : 1234  
age : 21

## 06\_getSessions.jsp

```
<h3>세션에 설정된 모든 값 얻어오기</h3>  
<%  
    Enumeration<String> names = session.getAttributeNames();  
    while (names.hasMoreElements()) {  
        String name = names.nextElement().toString();  
        String value = session.getAttribute(name).toString();  
        out.println(name + " : " + value + "<br>");  
    }  
%>
```

세션에 설정된 모든 값 얻어오기

id : james  
pwd : 1234  
age : 21

# 세션 객체의 여러 메소드 p.253

## 07\_infoSessions.jsp

<h3>세션 정보를 얻어오는 메소드를 사용하기</h3>

<%

```
String id_str = session.getId();
long lasttime = session.getLastAccessedTime();
long createdtime = session.getCreationTime();
long time_used = (lasttime - createdtime) / 1000;
int inactive = session.getMaxInactiveInterval() / 60;
boolean b_new = session.isNew();
```

%>

```
[1] 세션 ID는 [<%=session.getId()%>] 입니다.<br> <hr>
[2] 당신이 웹 사이트에 머문 시간은 <%=time_used%>초입니다.<br> <hr>
[3] 세션의 유효 시간은 <%=inactive%>분입니다.<br> <hr>
[4] 세션이 새로 만들어졌나요?<br>
```

<%

```
if (b_new) out.print(" 예! 새로운 세션을 만들었습니다.");
else out.print(" 아니오! 새로운 세션을 만들지 않았습니다.");
```

%>

세션 정보를 얻어오는 메소드를 사용하기

[1] 세션 ID는 [CF03631F0270F7C279B65C6610DFF664] 입니다.

[2] 당신이 웹 사이트에 머문 시간은 8초입니다.

[3] 세션의 유효 시간은 30분입니다.

[4] 세션이 새로 만들어졌나요?  
아니오! 새로운 세션을 만들지 않았습니다.

삭제 시에는 아래 두 가지 메소드 사용  
session.removeAttribute("age"); // age항목 제거  
session.invalidate(); // 세션 종료

# 1. 로그인 (login.html)

```
<body style="background-color:lime">
<div align="center">
  <h1>Campus Shopping Mall</h1>
  (car, bread, tea) <p>
  <form method="post" action="check_id.jsp">
    ID: <input type="text" name="id"/> <p>
    <input type="submit" value="login"/>
  </form>
  
</div>
</body>
```



*div*태그는 화면 구역을 나누는 태그입니다. *align* 속성은 정렬을 해줍니다.  
*img*태그는 그림을 불러옵니다. *alt*속성은 해당 그림 없을 때, 이 글을 대신 띄워줍니다.

"자카 웹툰 그림입니다. 저작권 어떻게 되나 모르겠는데, 상업적 용도 아니고 교육용 목적입니다~~"

## 2. 아이디 체크 (check\_id.jsp)

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("UTF-8");
    String id = request.getParameter("id");
    String[] members = {"a", "b", "c", "d", "james"};
    for(String m: members) {
        if (id.equals(m)){
            session.setAttribute("id", id);
            session.setMaxInactiveInterval(60*60); // 테스트 편의 위해 1시간
            response.sendRedirect("menu.jsp");
        }
    }
%>
<script>
    alert("미등록 아이디입니다~ 문의전화: 051)200-1035");
    location.href="login.html";
</script>
```

<script> ~ </script>는 자바스크립트입니다.  
alert은 팝업창입니다. location.href="~"는 다른 페이지로 이동합니다.

- 아이디를 세션에 저장 후 메뉴화면으로 자동 이동
- 특정 아이디가 아니면 팝업창 띄움

나중에는 데이터베이스 조회해서  
아이디 체크할 예정입니다.

### 3. 메뉴 (menu.jsp)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("UTF-8");
    String id = (String)session.getAttribute("id");
%>
<body style="background-color:yellow">
<h1>MENU</h1>
<hr>
<font color="blue"><%=id%></font>님, 오늘도 즐겁게 쇼핑하세요! <br>
<hr><p>
<ul>
    <li><a href = "">car</a>
    <li><a href = "">bread와 tea</a>
    <li><a href = "logout.jsp">logout</a>
</ul>
```

<font>태그는 글꼴을 의미합니다.  
<ul> ~ </ul>은 굵머리 기호. <ol> ~ </ol>은 번호매기기  
실제 항목들은 <li> 이용해서 하나하나 나열합니다.

## MENU

james님, 오늘도 즐겁게 쇼핑하세요!

- [car](#)
- [bread와 tea](#)
- [logout](#)

## 4. 로그아웃(logout.jsp)

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<% session.invalidate(); %>
<script>
    alert("로그아웃 되었습니다.");
    location.href="login.html";
</script>
```

- 세션 invalidate시킨 후 로그인 첫화면으로 이동

